Faster Fixed Parameter Tractable Algorithms for Finding Feedback Vertex Sets

VENKATESH RAMAN, SAKET SAURABH, AND C. R. SUBRAMANIAN

The Institute of Mathematical Sciences

Abstract. A feedback vertex set (*fvs*) of a graph is a set of vertices whose removal results in an acyclic graph. We show that if an undirected graph on n vertices with minimum degree at least 3 has a fvs on at most $\frac{1}{3}n^{1-\epsilon}$ vertices, then there is a cycle of length at most $\frac{6}{\epsilon}$ (for $\epsilon \geq 1/2$, we can even improve this to just 6).

Using this, we obtain a $O((\frac{12 \log k}{\log \log k} + 6)^k n^{\omega})$ algorithm for testing whether an undirected graph on n vertices has a fvs of size at most k. Here n^{ω} is the complexity of the best matrix multiplication algorithm. The previous best parameterized algorithm for this problem took $O((2k + 1)^k n^2)$ time.

We also investigate the fixed parameter complexity of weighted feedback vertex set problem in weighted undirected graphs.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Feedback vertex set, girth, parameterized complexity

1. *Introduction*

It is well known that a connected undirected graph possesses a cycle of logarithmic length if its minimum degree or average degree is at least 3. In this article, we obtain a similar result on the existence of short cycles in graphs having a small feedback vertex set (*fvs*). A feedback vertex set of a graph is a subset of vertices whose removal results in an acyclic graph. We show that if an undirected graph on n vertices with minimum degree at least 3 has a fvs on at most $\frac{1}{3}n^{1-\epsilon}$ vertices, then

---

Parts of this work were presented as extended abstracts in *Proceedings of 13th Annual International Symposium on Algorithms and Computation (ISAAC'02)*, Lecture Notes in Computer Science, vol. 2518, Springer-Verlag, New York, pp. 241–248, and *Proceedings of the 2nd Brazilian Symposium on Graphs, and Algorithms, and Combinatorics (GRACO)*, Electronic Notes in Discrete Mathematics, vol. 19, pp. 273–279.

Authors' address: Theoretical Computer Science, The Institute of Mathematical Sciences, CIT Campus, Taramani, Chennai, India, 600 113, e-mail: {vraman,saket,crs}@imsc.res.in.

there is a cycle of length at most $\frac{6}{\epsilon}$ (for $\epsilon \geq 1/2$, we can even improve this to just 6). This is one of the main contributions of this article.

Using this structural result, we also obtain a faster algorithm for solving the fvs problem. This problem is to determine, given an undirected graph G on n vertices and a nonnegative integer parameter k, whether G has a set of at most k vertices whose removal results in an acyclic graph. We explore efficient fixed parameter algorithms whose worst-case time complexity is bounded by functions of the form $f(k)n^{O(1)}$ where f is any function of k. Our focus is on making f as slowly growing (with k) as possible. Such an algorithm is quite useful in practice for small ranges of k (against a naive $n^{k+O(1)}$ algorithm). Problems for which such algorithms can be designed are said to be *Fixed Parameter Tractable (FPT)*. Several well-known NP-complete problems have been shown to be in FPT. Even if a problem is shown to be FPT, the quest for improving the function $f(k)$ goes on. For example, see Alber et al. [2001, 2002], Kanj and Perkovic [2002], and Fomin and Thilikos [2003] for recent attempts in improving the $f(k)$ in the case of the Planar Dominating Set problem. The book by Downey and Fellows [1999] provides a good introduction to the topic of Parameterized Complexity.

Fast parameterized algorithms have been proposed for the fvs problem earlier. See Downey and Fellows [1999] for a $O((2k+1)^k n^2)$ time algorithm. In a preliminary version of this article [Raman et al. 2002], the present authors substantially improved the dependence on k by developing a $O(\max\{12, 4\log k\}^k n^\omega)$[1] time deterministic algorithm. Recently, Kanj et al. [2004] have obtained an $O((2\log k + 2\log\log k + 18)^k n^2)$ time algorithm using results from extremal graph theory. If we are willing to accept uncertainty about the correctness of the answer, one can solve this problem in $O(4^k kn)$ time by a randomized algorithm presented in Becker et al. [2000].

We improve the known $f(k)$ function for this problem by developing an algorithm whose running time is $O((\frac{12\log k}{\log\log k} + 6)^k n^\omega)$ where ω is the exponent in the runtime for Matrix Multiplication.

All known algorithms for the problem use the bounded search tree technique. In particular, they work by finding a short cycle in the graph after some preprocessing, and branching recursively on each vertex of the short cycle. The improvement presented here also adheres to this paradigm. The correctness and efficiency of our algorithm is based on the new graph theoretical result (mentioned before) that connects minimum fvs size and the length of the shortest cycle.

Throughout the paper, we use $\delta(G)$ to denote the minimum degree of a graph G and $g(G)$ to denote its girth, that is, the length of a shortest cycle in the graph.

1.1. OUR RESULTS. First, we obtain the following result on short cycles.

THEOREM 1. *Let G be a graph on n vertices with minimum degree 3, having a feedback vertex set of size k, such that $(n-k) > 4 \cdot \binom{k}{2}$. Then, $g(G) \leq 6$.*

Based on Theorem 1, we present a simple FPT algorithm for finding a fvs of an undirected graph.

[1] All logarithms are to the base 2.

THEOREM 2. *Let G be an undirected graph on n vertices. Then, we can determine whether or not G has a feedback vertex set of size at most k (and find one if there is) in $O(\max\{6, 4\log k + 2\}^k n^{\omega})$ time.*

This is a substantial improvement over the $O((2k+1)^k n^2)$ time algorithm of Downey and Fellows [1999]. The Theorem 1 is further generalized as follows.

THEOREM 3. *Let $0 < \epsilon < 1$. Let G be a graph on n vertices such that (a) $\delta(G) \geq 3$, (b) $n \geq \lceil 3^{\frac{1}{\epsilon}} \rceil$, and (c) G has a fvs of size at most $\frac{1}{3} n^{1-\epsilon}$. Then G has a cycle of length less than $6/\epsilon$, that is, $g(G) < 6/\epsilon$.*

By applying Theorem 3 and by being more careful about analyzing and deciding whether we proceed by finding a short cycle or we apply brute-force approach, we obtain further significant improvements in the running time. Precisely, we show

THEOREM 4. *Let G be an undirected multigraph on n vertices. Then, we can determine whether or not G has a fvs of size at most k in time*

$$O\left(\left(\frac{12\log k}{\log\log k} + 6\right)^k n^{\omega}\right).$$

This results in an improvement in the running time by a factor of $(\frac{\log\log k}{6})^k$ over the result mentioned in Theorem 2.

We also show that the weighted feedback vertex set problem is solvable in essentially the same time if each vertex has real weight at least 1. In the general case, when the weights are arbitrary real numbers, we show that the problem is unlikely to be fixed parameter tractable.

1.2. ORGANIZATION OF THE REST OF THE ARTICLE. Given G and k, one can construct a graph G', in polynomial time, with $\delta(G') \geq 3$ such that G has a fvs of size at most k if and only if G' has a fvs of size at most k. Section 2 describes this preprocessing step. It also describes a generic, short cycle based branching algorithm that will be used by the rest of the algorithms we develop.

Section 3 proves one of the main structural results, namely, Theorem 1. It also derives Theorem 2 as a consequence. Section 4 presents the proof of the other structural result, Theorem 3 and presents a description of the algorithm stated in Theorem 4.

Section 5 investigates the parameterized complexity of weighted feedback vertex set. Finally Section 6 concludes with remarks and further directions.

Throughout this article, we use $G = (V, E)$ to denote an undirected (multi)-graph on n vertices and m edges. For a subset $S \subseteq V$, $G[S]$ is the induced subgraph of G on S. We assume that all graphs are represented in the form of adjacency lists.

2. *Preliminaries*

In this section, we first describe some preprocessing that removes non-essential vertices from the input graph without affecting the size of minimum fvs. Then, we present a generic algorithm for finding a fvs which is going to be the template of our main algorithmic results.

2.1. PREPROCESSING. The following lemma is not difficult to verify and is well known in the literature on fvs problems. See, for example, Bar-Yehuda et al. [1998] for proofs.

LEMMA 1. *Let G be an undirected multigraph. Perform the following steps as long as possible.*

(1) *If G has a vertex of degree ≤ 1, remove it (along with the incident edge if any).*
(2) *If G has a vertex x of degree 2 adjacent to vertices y and z, $y \neq x$ and $z \neq x$, short circuit by removing x and joining and y and z by a new edge (even if y and z were adjacent earlier).*

Let G' be the resulting multigraph. Then G has a feedback vertex set of size at most k if and only G' has a feedback vertex set of size at most k.

Clearly, the graph G' is such that each component of G' has minimum degree at least three unless that component is *either* an empty graph *or* a graph on one vertex with a self loop (in which case that component has a feedback vertex set of size 1). G' can be constructed in $O(m)$ steps where m is the number of edges in G.

LEMMA 2 [BAR-YEHUDA ET AL. 1998]. *Given an undirected multi graph $G = (V, E)$ on m edges, in $O(m)$ time we can produce a multigraph G' with minimum degree 3 such that G has a fvs of size k if and only if G' has a fvs of size k.*

We also note that:

LEMMA 3. *Given an undirected multigraph $G = (V, E)$ on n vertices with $\delta(G) \geq 3$, removing a vertex v from G can be achieved in $O(n)$ time.*

We recall the following well-known algorithmic results:

LEMMA 4 [CORMEN ET AL. 2001]. *Given an undirected multigraph G, we can test whether G has a cycle or not in $O(n)$ time, where n is the number of vertices in G.*

LEMMA 5 [ITAI AND RODEH 1978]. *Given an undirected multigraph G, a shortest cycle (if there is any) in G can be found in $O(\min\{mn, n^{\omega}\})$ time where n^{ω} is the running time of the best-known algorithm for multiplying two n by n matrices.*

LEMMA 6 [ITAI AND RODEH 1978]. *Given an undirected graph G on n vertices, a cycle of length at most $g(G) + 1$ in G can be found in $O(n^2)$ time.*

2.2. A GENERIC ALGORITHM. The following generic algorithm forms the basis of our main algorithmic results. Here, G is an undirected multigraph and $k \geq 0$. The algorithm returns YES and a feedback vertex set of size at most k in G if there is one and returns NO otherwise.

Algorithm GFBVS(G, k)

—***Step* 0′:** If G is acyclic, then answer YES and return $\emptyset$.

—***Step* 0:** If $k = 0$ and G contains a cycle, then answer NO and EXIT.

—***Step* 1:** Apply Lemma 1 to get G'.

—***Step* 2:** Find a shortest cycle C in G'. (C could possibly be of length 1 or 2.)

—***Step* 3:** If for some vertex $v \in C$, FBVS($G' - v, k - 1$) is true, then answer YES and return $\{v\} \cup$ FBVS($G' - v, k - 1$), else answer NO.

The correctness of the algorithm follows from Lemma 1 and the fact that any feedback vertex set must contain a vertex from every cycle in the graph.

Furthermore, if $g(G') \leq g$ for all the graphs G' used in Step 2 of the recursive calls, then the overall algorithm takes $O(g^k n^\omega)$ time. This is because the recursion tree at Step 3 has a branching factor of at most g and depth at most k, and Step 2 takes $O(n^\omega)$ time from Lemma 5. Steps 0 and 1 each takes $O(m)$ time by Lemma 4. Also, in Step 2, instead of finding a shortest cycle, if we find a cycle of length at most $g + 1$, then by Lemma 6, the running time of the algorithm will be $O((g + 1)^k n^2)$ time. Thus we have

LEMMA 7. *Let G be an undirected graph, and let g be the maximum size of the girth of the graphs G' used in Step 2 of* **GFBVS**(G, k)*. Then we can find a feedback vertex set of size at most k in G (or determine its absence) in $O(g^k n^\omega)$ time or in $O((g + 1)^k n^2)$ time.*

Erdös and Posa [1962] observed that girth of any undirected graph G with minimum degree ≥ 3 is bounded by $2 \log n$. Given such a graph, one can find in $O(n)$ time a cycle of length at most $2 \log n$ by growing a Breadth First Search (BFS) tree till the first non-tree edge is encountered. Thus, we get

LEMMA 8. *Any graph G with minimum degree at least 3 has a cycle of length at most $2 \log n$ and such a cycle can be found in $O(n)$ time where n is the number of vertices in the graph G.*

So, in Step 2 of the generic algorithm if we find just a cycle of length at most $2 \log n$ (which may not necessarily be the shortest cycle), then from Lemma 8 and Lemma 7, we have

THEOREM 5. *Given a graph G on n vertices, and an integer parameter k, we can determine whether or not G has a feedback vertex set of size at most k in $O((2 \lg n)^k n + m)$ time, or in $O((4k \log k)^k n + nm)$ time.*

The second bound follows from the observation that

$$(2 \log n)^k \leq (4k \log k)^k + n$$

for all n and $k \leq n$.

3. *Proofs of Theorems* 1 *and* 2

We first present the proof of Theorem 1 and discuss the tightness of our result. Then we derive Theorem 2 as a consequence. The arguments in Theorem 1 are based on the following lemma.

LEMMA 9. *Let $T = (V, E)$ be a forest on N vertices. Let $M' = \{(i, j) \in E \mid deg_T(i) = deg_T(j) = 2\}$ and $L = \{a \in V \mid deg_T(a) \leq 1\}$. Then there exist $M \subseteq M'$, such that M is a matching and $|W = L \cup M| \geq N/4$.*

PROOF. Without loss of generality assume that T is a tree, otherwise we can apply the result on each tree of the forest and combine them to get the result. Now let v be a vertex of degree not equal to 2. Root the tree at v and direct the edges away from the root, call it T_v. Let D_2 be the set of degree 2 vertices of T and $D_{\geq 3}$ denotes the set of vertices of degree at least 3 in T. Every connected component of

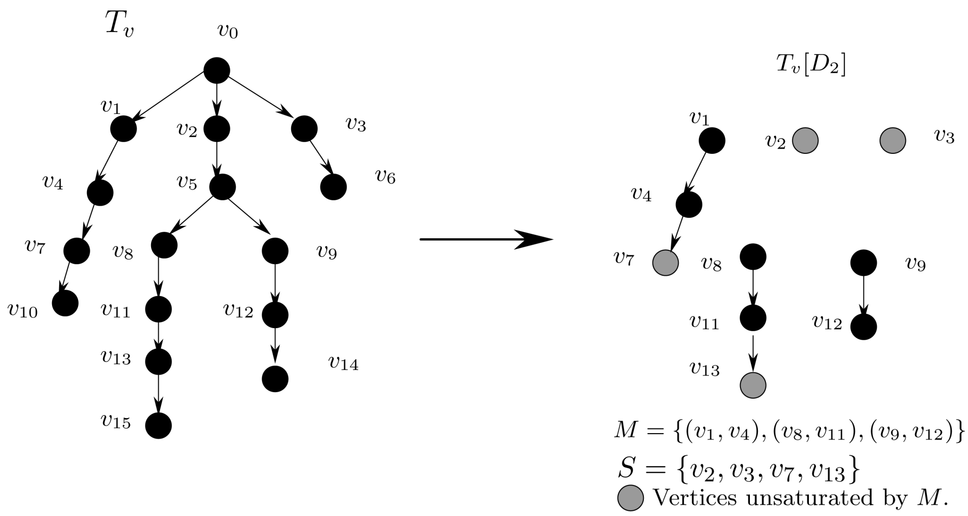

FIG. 1. Illustration for Lemma 9.

the induced graph $T_v[D_2]$ is either an isolated vertex or a directed path. Let M be a matching on degree 2 vertices consisting of all the alternate edges starting from the vertices of indegree 0 of all the paths of $T_v[D_2]$ ignoring their direction. Clearly $M \subseteq M'$. Define S to be the set of vertices in D_2 unmatched by any edge in M. Note that each vertex in S is either an isolated vertex of $T_v[D_2]$ or the last vertex in an even length directed path of $T_v[D_2]$. See Figure 1. Now with each vertex $u \in S$, associate its unique child in T_v, whose degree is either 1 or at least 3 in T. Note that this association is injective, as in a rooted tree we have a unique parent for every vertex other than root. This implies that $|S| \leq |L| + |D_{\geq 3}|$. It is well known that the number of vertices of degree at least 3 in a tree is smaller than the number of leaves of the tree. So, we have $|D_{\geq 3}| < |L|$ and $|S| \leq |L| + |D_{\geq 3}| \leq 2|L|$. This gives

$$N = |L| + 2|M| + |S| + |D_{\geq 3}| < |L| + 2|M| + 2|L| + |L| \leq 4|L| + 2|M|.$$

Dividing both sides by 4 gives $N/4 < |L| + |M|/2 \leq |W = L \cup M|$. This completes the proof. □

PROOF OF THEOREM 1. Without loss of generality, we assume that G is a simple graph having no self-loops and parallel edges. Let F be a fvs of size k in G and let T denote the induced forest $G[V - F]$ on $N = n - k$ vertices. Apply Lemma 9 to T to get W (mentioned in the Lemma 9). Now with every element $a \in W$, we associate a (unordered) pair of vertices of F as follows:

Case 1. $a \in L$, that is, a is a vertex of degree 0 or 1.
Since degree of a is at least 3 in G, a has at least two neighbors in F. We pick arbitrarily two of these neighbors and associate them with the a. We use $\{x_a, y_a\}$ to denote the pair associated with vertex a.

Case 2. $a = (u, v)$ is an edge from M. Since each of u and v has degree at least 3 in G, each of them has at least one neighbor in F. We pick one neighbor (in F) of each u and v and associate them with a. We use $\{x_u, x_v\}$ to denote the pair associated with $a = (u, v)$. Note that $x_u = x_v$ possibly.

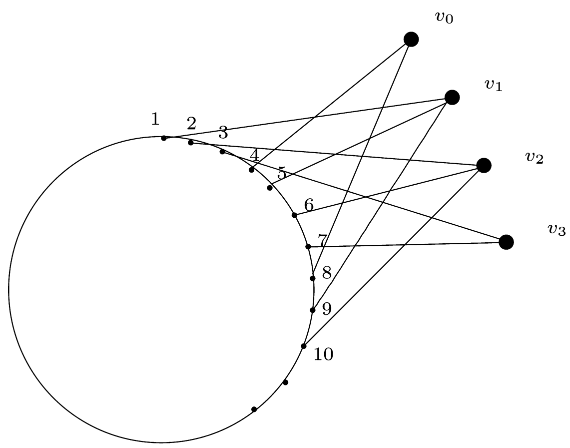

FIG. 2. Graph G with $g(G) = 6$, $|FVS| = 5$ and $\delta(G) \geq 3$.

Suppose there is a pair $\{x, y\}$ associated with some $a \in W$ such that $x = y$. In that case, a should be an edge $(u, v) \in M$. Thus, we get a 3-cycle (x, u, v, x) in G proving that $g(G) \leq 6$. Hence, we assume that every selected pair $\{x, y\}$ is such that $x \neq y$. Since the number of such pairs is at most $\binom{k}{2}$ and as $|W| \geq \frac{N}{4} > \binom{k}{2}$, the association map is not injective. That is, there are $a_1, a_2 \in W$, $a_1 \neq a_2$ such that both a_1 and a_2 are associated with some pair $\{x, y\}$ with $x \neq y$. The following cases arise.

—Both a_1 and a_2 are vertices of degree at most 1. In that case, we get a 4-cycle (x, a_1, y, a_2, x), thereby proving that $g(G) \leq 4$.

—Both a_1 and a_2 are edges (u_1, v_1) and (u_2, v_2) from M such that x is a neighbor of u_1 and u_2 and y is a neighbor of v_1 and v_2. In this case, we get a cycle $(u_1, x, u_2, v_2, y, v_1, u_1)$ of length 6. This leads to a cycle of length 6, thereby proving that $g(G) \leq 6$.

—Without loss of generality, a_1 is a vertex a of degree ≤ 1 and a_2 is an edge (u, v) from M. Also, x is a neighbor of a and u and y is a neighbor of a and v. This gives rise to a cycle (a, x, u, v, y, a) of length 5, proving again that $g(G) \leq 6$.

In any case, we are guaranteed to have a cycle of length at most 6 thereby proving Theorem 1. □

COROLLARY 1. *If a graph on n vertices with minimum degree* 3 *has a feedback vertex set of size at most* $\sqrt{n/2}$, *then* $g(G) \leq 6$.

Remark 1. Corollary 1 is tight in the sense that there are graphs G satisfying the hypothesis of the corollary with $g(G) = 6$, as the following example (Figure 2) shows.

Let $G = (V, E)$ be a graph on $n \geq 63$ vertices such that n is a multiple of 4. The graph has a cycle C of length $n - 4$ on vertices 1 to $n - 4$. The remaining 4 vertices are named v_0, v_1, v_2 and v_3. The vertex v_i is adjacent to all vertices j in the cycle such that $j \bmod 4 = i$.

It is easy to see that $g(G) = 6$, and $|F| \leq 5$ as $\{1, v_0, v_1, v_2, v_3\}$ is a feedback vertex set of G.

COROLLARY 2. *Let G be a graph on n vertices with minimum degree 3 having a feedback vertex set of size at most k. Then,* $g(G) \leq \max\{6, 4\lg k + 2\}$.

PROOF. If $k \leq \sqrt{n/2}$, then $g \leq 6$ by the previous corollary. Otherwise, $n \leq 2k^2$ and hence, by Lemma 8, we have $g \leq 2\log n \leq 4\log k + 2$. □

PROOF OF THEOREM 2. Modify the Step 2 in our generic algorithm as follows:

—***Modified Step 2.*** Find a shortest cycle C in G'. If $k \leq \sqrt{n/2}$ and $g > 6$, then answer NO.

Now use Corollary 2 and apply Lemma 7 over **GFBVS**(G, k) with Modified Step 2 to get Theorem 2. □

4. *Proofs of Theorems* 3 *and* 4

We can generalize Theorem 1 and Corollary 1 to prove upper bound for girth in graphs having larger sized feedback vertex set (than assumed in Theorem 1). We will need the following result of Alon et al. [2002].

THEOREM 6 [ALON ET AL. 2002]. *Any graph $G = (V, E)$ on n vertices with average degree d, contains a cycle of length* $\leq 2\log_{d-1} n + 2$.

Using this result, we prove Theorem 3.

PROOF OF THEOREM 3. We can assume that $\epsilon < 1/2$ as otherwise the theorem follows from Corollary 1. Let G be a graph on $n \geq n_0 = \lceil 3^{1/\epsilon} \rceil$ vertices with minimum degree 3 and having a feedback vertex set F of size $k \leq n^{1-\epsilon}/3$. As before, let T denote the induced forest on the remaining $N = n - k$ vertices in G.

We construct a new multigraph G' with $V(G') = F$ as follows. The edges of G' are included as follows. For every $a \in W$ (where W is the set obtained by applying Lemma 9 on $T = G[V - F]$), we include an edge between x_a and y_a ($x_a = y_a$ possibly) where $\{x_a, y_a\}$ is the the pair associated with a in F in the proof of Theorem 1. By Lemma 9, we know that W is of size at least $N/4$. It follows that G' has at least $N/4$ edges and hence its average degree is $\geq N/2k$ as $|V'| = k$.

Note that if G' has a cycle of length at most g, then G has a cycle of length at most $3g$, as any edge of the cycle in G' can be replaced by a path of length at most 3 in the original graph G. It is possible that G' has either a self-loop or two parallel edges joining the same pair of vertices in F. In that case, by way of construction, a self-loop or two parallel edges lead to a cycle of length at most 6 in G, thereby showing that $g(G) \leq 6 \leq 6/\epsilon$. Hence we assume, without loss of generality, that G' is a simple graph with average degree at least $N/2k$.

By Theorem 6, G' has a cycle of length at most

$$(2\log_{(N/2k)-1} k) + 2 = \frac{2\log k}{\log((N/2k) - 1)} + 2.$$

This implies that G has a cycle of length at most

$$\frac{6\log k}{\log((N/2k) - 1)} + 6.$$

After substituting $N = n - k$ and $k \leq \frac{1}{3}n^{1-\epsilon}$, we get

$$\begin{aligned} g(G) &\leq \frac{6\log(n^{1-\epsilon}/3)}{\log\left(\frac{3}{2}n^{\epsilon} - \frac{3}{2}\right)} + 6 \\ &< \frac{6(1-\epsilon)\log n}{\log n^{\epsilon}} + 6 \qquad \text{for } n \geq n_0. \\ &\leq \frac{6(1-\epsilon)}{\epsilon} + 6 = \frac{6}{\epsilon} \end{aligned}$$

which is what we wanted to show. □

COROLLARY 3. *Let G be a graph on n vertices with minimum degree ≥ 3 having a feedback vertex set of size at most k. Then, for every ϵ $(0 < \epsilon < 1)$ such that $n \geq \lceil 3^{1/\epsilon} \rceil$, $g(G) \leq \max\{\frac{6}{\epsilon}, \frac{2\log 3k}{1-\epsilon}\}$.*

PROOF. If $k \leq n^{1-\epsilon}/3$, then $g \leq 6/\epsilon$ by the previous theorem. Otherwise, $n \leq (3k)^{\frac{1}{1-\epsilon}}$ and hence, by Lemma 8, $g \leq \frac{2\log 3k}{1-\epsilon}$. □

For fixed values of ϵ, the lower bound on n (required to apply Corollary 3) is also fixed. Hence, by applying Lemma 7 to **GFBVS**(G, k), with modified step 2, very similar to the one in the previous section, we get the following

THEOREM 7. *Let G be an undirected graph on n vertices. Then, for every fixed ϵ, $0 < \epsilon < 1$, we can determine whether or not G has a fvs of size at most k, and find one if it exists, in $O(\max(\frac{6}{\epsilon}, \frac{2\log 3k}{1-\epsilon})^k n^{\omega})$ time.*

4.1. A FASTER ALGORITHM. In this section, we prove Theorem 4 by providing a faster FPT algorithm for undirected fvs problem. By the phrase, *kernel of the problem*, we mean an equivalent reduced instance I' of the original problem instance I, where the size of I' is bounded by some function of the parameter k. Note that Theorem 3 gives us a kernel of size $(3k)^{1/1-\epsilon}$ for every ϵ, $0 < \epsilon < 1$, in time $O((\frac{6}{\epsilon})^k n^{\omega})$. The new algorithm makes use of this reduction by choosing a proper ϵ and obtains a kernel. After that, it works by a brute-force approach instead of branching on a short cycle. The algorithm is presented below. As usual, G is an undirected multigraph and $k \geq 0$ is an integer.

Algorithm Mod-FBVS(G, k)

—***Step* 0 :** If G is acyclic answer YES or if $k = 0$ answer NO.

—***Step* 1 :** Apply Lemma 1 to get G'.

—***Step* 2:** Find a shortest cycle C in G'. Let g be its length.

—***Step* 3(*a*):** If $g < 6(\frac{\log k + \log\sqrt{\log k}}{\log\sqrt{\log k}})$, then if for some $v \in C$, Mod-FBVS$(G' - v, k - 1)$ is true then answer YES and return $\{v\} \cup$ Mod-FBVS$(G' - v, k - 1)$, else answer NO.

—***Step* 3(*b*):** Try all possible k-subsets of $V(G)$ as a possible feedback vertex set of G and say YES, if any such subset is a fvs and return that subset, else say NO.

Correctness of the algorithm Mod-FBVS follows from its description. When we reach Step 3(b) of the algorithm, we have $g \geq 6(\frac{\log k + \log\sqrt{\log k}}{\log\sqrt{\log k}})$. Then, we use Theorem 3 by choosing $\epsilon = \frac{\log\sqrt{\log k}}{\log k + \log\sqrt{\log k}}$ and observing that $n \geq \lceil 3^{1/\epsilon} \rceil$ since by

Lemma 8,

$$\begin{aligned} 2\log n \;&\geq\; g \;\; \geq 6\left(\frac{\log k + \log\sqrt{\log k}}{\log\sqrt{\log k}}\right) \\ \log n \;&\geq\; \frac{3}{\epsilon} \\ n \;&\geq\; 8^{\frac{1}{\epsilon}} \;\;\geq\;\; \lceil 3^{\frac{1}{\epsilon}} \rceil. \end{aligned}$$

Thus, we have

$$\frac{1}{3}n^{1-\epsilon} < k \iff n < (3k)^{\frac{1}{1-\epsilon}} \iff n < (3k)^{\frac{\log k + \log\sqrt{\log k}}{\log k}} \leq 9k\sqrt{\log k}$$

So either the girth is bounded by $\frac{12\log k}{\log\log k} + 6$ or we have a kernel of size $\leq 9k\sqrt{\log k}$. So the time complexity of the algorithm is bounded by:

$$\max\left\{\left(\frac{12\log k}{\log\log k} + 6\right)^k n^{\omega}, \binom{9k\sqrt{\log k}}{k} n^2 \sim (9e\sqrt{\log k})^k n^2\right\}.$$

Since the first function is asymptotically bigger, we use it to bound the time complexity. Combining all these, we complete the proof of Theorem 4.

5. *Weighted Feedback Vertex Set*

The WEIGHTED FEEDBACK VERTEX SET problem (*WFVS* for short) is: given an undirected graph $G = (V, E)$, a weight function $w : V \to \mathbf{R}^+$, and $k \in \mathbf{R}^+$, find a feedback vertex set F with total weight at most k. The weight of F is defined as the sum of weights of $v \in F$.

In the weighted case, the preprocessing described in Lemma 1 cannot be applied as such because it is possible that every minimum weight fvs contains some degree two vertex. However, if we assume that $w(v) \geq 1$ for every v, then we can modify the preprocessing as follows. Given a graph G with a vertex weight function w, we repeatedly remove vertices of degree 1 to transform G into G'' with minimum degree ≥ 2. Then, for every path P in G'' joining two vertices x and y of larger (≥ 3) degrees such that each internal vertex of P has degree two, we replace P by the path xzy where z is an internal vertex of P having minimum weight among all internal vertices of P. Let G' be the resulting weighted graph. The weights of vertices surviving in G' are the same assigned to them in G.

Now it is easy to verify that G has a feedback vertex set of weight at most k if and only if G' has a fvs of weight at most k. Let us call such a graph having minimum degree 2 with each degree 2 vertex connected to two vertices of larger degree as a *branchy graph*. It is not difficult to adapt Theorem 3 for branchy graphs and obtain its weighted version that shows that if a weighted branchy graph G has a fvs of weight at most $\frac{1}{3}n^{1-\epsilon}$, where $0 < \epsilon < 1$, then $g(G) < \frac{12}{\epsilon}$, provided $n \geq \lceil 3^{\frac{1}{\epsilon}} \rceil$.

For the weighted case, we modify the algorithm Mod-FBVS by reducing G to a branchy graph (as described before) in Step 1. We then look for a cycle of length

$$g < 12\left(\frac{\log k + \log\sqrt{\log k}}{\log\sqrt{\log k}}\right)$$

in Step 3(a) of the algorithm. As before, the algorithm either finds a short cycle and branches on the vertices of the cycle or applies brute force. In the first case, since each vertex picked for branching has weight at least 1, the depth of the recursion is at most k. Also, we can show that in Step 3(b), $n = O(k\sqrt{\log k})$, as in the unweighted case by using the fact that every vertex has weight at least 1. Thus, we have an analogue of Theorem 4

THEOREM 8. *Given an undirected graph $G = (V, E)$, a positive real parameter k and a weight function w from V to $\mathbf{R}^+$ such that for every $v \in V$, $w(v) \geq 1$, we can determine whether or not G has a feedback vertex set of weight at most k in time*

$$O\left(\left(\frac{24\log k}{\log\log k} + 12\right)^k n^\omega\right).$$

General-WFVS is the problem of finding a fvs of weight at most k, when the weights of the vertices are arbitrary real numbers. We show that the problem is not fixed parameter tractable unless $P = NP$ by proving that it is *NP*-complete for any fixed $k > 0$. We can give a direct reduction from the *NP*-complete, unweighted fvs problem on undirected graphs to General-WFVS with $k = 1$, by defining the weight function w to be $w(v) = 1/k$ for all $v \in V$. In fact this implies that there cannot be a $f(k)n^{O(1)}$ or even $n^{O(k)}$ time algorithm for General-WFVS problem unless $P = NP$.

THEOREM 9. *General-WFVS problem is not fixed parameter tractable unless $P = NP$.*

6. *Conclusions and Further Work*

In this article, we proved that graphs with minimum degree 3 having a small fvs possess short cycles. Using this, we obtained faster algorithms for parameterized feedback vertex set problem on undirected graphs. Our main result achieves a significant improvement in the dependence on k (the parameter) of the running time. We get an algorithm with $O((\frac{12\log k}{\log\log k} + 6)^k n^\omega)$ running time.

Based on the preliminary report of our work, a number of advances have been made on reducing the $f(k)$ for the fvs algorithm. Dehne et al. [2005] have recently obtained an algorithm for fvs that runs in time $O(c^k n^3)$, where $c = 10.567$. Independently, Guo et al. [2005] obtained an $O(c^k mn)$ time algorithm for the fvs problem, where $c = 37.7$.

Apart from its application to the design of FPT algorithms, our Lemma 9 and Theorems 1 and 3 may be of independent interest in extremal graph theory. Also, all of the recent improvements on designing FPT algorithms are based (in part) directly or indirectly on the ideas used in this article.

Theorem 3 essentially shows that the size of the problem kernel for the feedback vertex set problem is $O(k^{1+2\epsilon})$ for fixed $\epsilon \leq 1/2$. This is because we can reduce the problem size to $O(k^{1+2\epsilon})$ in $O((\frac{6}{\epsilon})^k n^\omega)$ time. In particular, when $\epsilon = 1/2$, Corollary 1 gives us a kernel of size $O(k^2)$ in $O(6^k n^\omega)$ time. We can use this kernel in connection with newly developed algorithms by first branching on cycles of length at most 6 and using the improved c^k algorithms only when girth of the graph exceeds 6 in which case the instance size is at most $O(k^2)$. This will give a fixed parameter tractable algorithm with time complexity $O(6^k n^\omega + (10.567)^k k^6)$ using the algorithm developed in Dehne et al. [2005]. Following questions are interesting and still remain unanswered.

—Can we get a polynomial size kernel for feedback vertex set in polynomial time?[2]
—Is the feedback vertex set problem fixed parameter tractable in directed graphs?
—Is the feedback vertex set problem fixed parameter tractable even in planar directed graphs?

ACKNOWLEDGMENTS. We thank R. Balasubramanian, Mike Fellows, Vinod Parthasarathy and P. Paulraja for fruitful discussions and comments on earlier drafts of the article, and Santosh Vempala for suggesting the direction for generalization in Theorem 3. We thank the anonymous referees for several useful comments; the version of the proof of Lemma 9 in this article is due to one of them and is simpler than our earlier proof.

REFERENCES

ALBER, J., BODLAENDER, H. L., FERNAU, H., KLOKS, T., AND NIEDERMEIER, R. 2002. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica 33*, 4, 461–493.

ALBER, J., FAN, H., FELLOWS, M. R., FERNAU, H., NIEDERMEIER, R., ROSAMOND, F. A., AND STEGE, U. 2001. Refined search tree technique for dominating set on planar graphs. In *Proceedings of 26th International Symposium on Mathematical Foundations of Computer Science (MFCS)*. Lecture Notes in Computer Science, vol. 2186. Springer-Verlag, New York, 111–122.

ALON, N., HORRAY, S., AND LINIAL, N. 2002. The moore bound for irregular graphs. *Graphs Combinat. 18*, 53–57.

BAR-YEHUDA, R., GEIGER, D., NAOR, J., AND ROTH, R. M. 1998. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and bayesian inference. *SIAM J. Comput. 27*, 4, 942–959.

BECKER, A., BAR-YEHUDA, R., AND GEIGER, D. 2000. Randomized algorithms for the loop cutset problem. *J. Artif. Intel. Research 12*, 219–234.

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. 2001. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill, New York.

DEHNE, F. K. H. A., FELLOWS, M. R., LANGSTON, M. A., ROSAMOND, F. A., AND STEVENS, K. 2005. An $O(2^{O(k)} n^3)$ FPT algorithm for the undirected feedback vertex set problem. In *Proceedings of 11th International Computing and Combinatorics Conference (COCOON)*. Lecture Notes in Computer Science, vol. 3595. Springer-Verlag, New York, 859–869.

DOWNEY, R., AND FELLOWS, M. 1999. *Parameterized Complexity.* Springer-Verlag, New York.

ERDÖS, P., AND POSA, L. 1962. On the maximal number of disjoint circuits of a graph. *Publ. Math. Debrecen 9*, 3–12.

ESTIVILL-CASTRO, V., FELLOWS, M. R., LANGSTON, M. A., AND ROSAMOND, F. A. 2006. The undirected feedback vertex set problem has a poly(k) kernel. In *Proceedings of 2nd International Workshop on Parameterized and Exact Computation (IWPEC)*. Lecture Notes in Computer Science, Springer-Verlag, New York.

FOMIN, F. V., AND THILIKOS, D. M. 2003. Dominating sets in planar graphs: branch-width and exponential speed-up. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, New York, 168–177.

GUO, J., GRAMM, J., HÜFFNER, F., NIEDERMEIER, R., AND WERNICKE, S. 2005. Improved fixed-parameter algorithms for two feedback set problems. In *Proceedings of 9th Workshop on Algorithms and Data Structures (WADS)*. Lecture Notes in Computer Science, vol. 3608. Springer-Verlag, New York, 158–168.

[2]Recently, Estivill-Castro et al. [2006] have obtained an $O(k^{11})$ size kernel for the parameterized feedback vertex set problem in polynomial time.

ITAI, A., AND RODEH, M. 1978. Finding a minimum circuit in a graph. *SIAM J. Comput. 7*, 4, 413–423.

KANJ, I. A., PELSMAJER, M. J., AND SCHAEFER, M. 2004. Parameterized algorithms for feedback vertex set. In *Proceedings of 1st International Workshop on Parameterized and Exact Computation (IWPEC)*. Lecture Notes in Computer Science, vol. 3162. Springer-Verlag, New York, 235–247.

KANJ, I. A., AND PERKOVIC, L. 2002. Improved parameterized algorithms for planar dominating set. In *Proceedings of 27th International Symposium on Mathematical Foundations of Computer Science (MFCS)*. Lecture Notes in Computer Science, vol. 2420. Springer-Verlag, New York, 399–410.

RAMAN, V., SAURABH, S., AND SUBRAMANIAN, C. R. 2002. Faster fixed parameter tractable algorithms for undirected feedback vertex set. In *Proceedings of 13th Annual International Symposium on Algorithms and Computation (ISAAC)*. Lecture Notes in Computer Science, vol. 2518. Springer-Verlag, New York, 241–248.

RAMAN, V., SAURABH, S., AND SUBRAMANIAN, C. R. 2005. Faster algorithms for feedback vertex set. In *Proceedings of 2nd Brazilian Symposium on Graphs, Algorithms, and Combinatorics (GRACO)*. Electronic Notes in Discrete Mathematics, vol. 19. 273–279.

RECEIVED NOVEMBER 2004; REVISED JUNE 2005 AND FEBRUARY 2006; ACCEPTED FEBRUARY 2006